

# Конечные автоматы в разработке программных систем

Смирнова Н.В, Смирнов В.В, Кировоградский национальный технический университет,  
swckntu@rambler.ru

*The problem to eliminate ambiguities in the finite state machine when creating control software systems solution. Created a block diagram of a finite automaton model with the time to wait for the results of functions and their implementation, counting the events number and states stack.*

## ВВЕДЕНИЕ

В настоящее время расширяется область применения SWITCH-технологии [1] в разработке программного обеспечения систем управления различными объектами. В основе SWITCH-технологии лежит теория конечных автоматов (Finite State Machine).

Системы на основе конечных автоматов обладают детерминированным поведением, однозначными результатами тестирования, простой диагностикой, документированием и сопровождением на протяжении всего жизненного цикла системы. Для использования SWITCH-технологии в разработке программного обеспечения необходимо устранить ряд ограничений, сдерживающих ее широкое применение.

## СОЗДАНИЕ МОДЕЛИ АВТОМАТА

Современные языки программирования являются объектно-ориентированными и реализуют событийную модель для создания программного обеспечения, которая хорошо согласуется с основными положениями SWITCH-технологии.

Тем не менее, ни один язык программирования общего применения не содержит в себе никаких специальных средств для поддержки SWITCH-технологии.

Поэтому создание автоматных шаблонов проектирования позволит в полной мере реализовать преимущества данной техноло-

гии. В области создания управляющих программных систем получили системы на базе конечного автомата Мура [2], работа которого описывается выражением:

$$\begin{aligned} a(t+1) &= f[a(t), x(t)], \\ y(t) &= f(a(t)), \end{aligned} \quad (1)$$

где:  $a(t+1)$  – состояние автомата в момент времени  $t+1$ ,  $a(t)$  – состояние автомата в момент времени  $t$ ,  $x(t)$  – входное воздействие (событие) на автомат в момент времени  $t$ ,  $y(t)$  – управляющее воздействие (action) соответствующее выходной функции автомата в состоянии  $a$  в момент времени  $t$ .

На рис.1 представлена программная модель автомата Мура, в которую входят следующие переменные:  $x$  – для фиксации входного воздействия  $X$ ,  $a$  – для фиксации текущего состояния автомата  $A$ ,  $y$  – для указателя на функцию  $f$  управляющего воздействия  $Y$ , вызываемую в текущем состоянии  $a$  автомата:

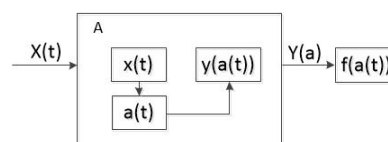


Рисунок 1 – Программная модель автомата Мура

Анализ модели показывает, что программная система, имеющая в своей основе данную модель, является нежизнеспособной, поскольку не учитывает целый ряд факторов, присущих реальным объектам управления и требований, предъявляемых к управляющим программам. Например, ожидание завершения выполнения функции

$f(a)$  приводит к «зависанию» программы и неспособности системы реагировать на входные воздействия. Решение проблемы потери управляемости системы во время ожидания завершения функции  $f(a)$  может иметь два варианта решения:

1. Выход из состояния ожидания по истечению интервала тайм-аута.
2. Запуск функции  $f(a)$  в параллельном потоке [3] с использованием тайм-аута.

Необходимым условием правильной работы автомата и является оценка результата выполнения функции  $f(a)$ , то есть, функция по своему завершению должна возбуждать внешнее событие  $R$ , включающее в себя результат завершения.

Следующей проблемой является неоднозначность при возврате автомата из состояния  $a(t)$  в состояние  $a(t-1, t-2 \dots t-n)$  при одинаковых событиях  $X0$  (рис.2).

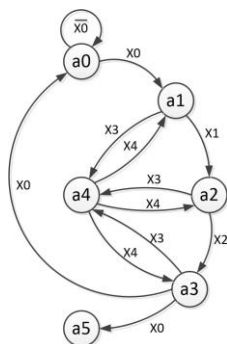


Рисунок 2 – Неоднозначность при возврате автомата в предыдущее состояние

Решением задачи устранения этой неоднозначности является введение в модель автомата стека состояний  $S$ .

Еще одна неоднозначность возникает в состоянии  $a3$  при поступлении события  $X0$ , когда необходимо осуществить переход автомата в процессе выполнения цикла (рис. 2). Если цикл не завершен, то должен произойти переход автомата из состояния  $a3$  в состояние  $a0$ , в противном случае – переход в состояние  $a5$  для завершения работы.

Для корректного поведения модели автомата, в нее введен счетчик циклов  $C$ . Таким образом, учитывая все дополнения, выражение (1) примет вид:

$$a(t+1) = f[a(t), x(t), To(f(a)), R(f(a)), S(a(i)), C(x(j))] \quad (2)$$

$$y(t) = f(a(t)).$$

С учетом полученного выражения (2) модель конечного автомата для создания шаблона базового класса будет иметь следующую структуру (рис. 3):

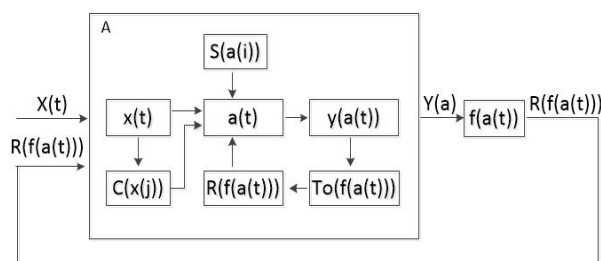


Рисунок 3 – Структура модели конечного автомата для реализации базового класса

Полученная структура модели конечного автомата легко реализуема в терминах языков программирования C++, Java и других языков программирования.

## Выводы

Реализация базового класса конечного автомата на основе созданной модели и результаты работы тестовой программы позволяет сделать выводы о жизнеспособности и достаточной эффективности управляющих программ на базе SWITCH-технологии.

## ЛИТЕРАТУРА

1. Шальто А. А. Switch-технология. Алгоритмизация и программирование задач логического управления / А. А. Шальто/ - СПб.: Наука, 1998. – 628 с.
2. Шальто А. А. Автоматное программирование / А. А. Шальто, Н. И. Поликарпова. - СПб.: Питер, 2011. - 176 с.
3. Татарчевский В. Применение Switch-технологии при разработке прикладного программного обеспечения для микроконтроллеров / В. Татарчевский. - Компоненты и технологии №11, №12 - 2006, №1, №2- 2007.